COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE 114 (2014) 70–79

journal homepage: www.intl.elsevierhealth.com/journals/cmpb

A simple versatile solution for collecting multidimensional clinical data based on the CakePHP web application framework

Martin Biermann[a,b,*]

[a] *Nuclear Medicine and PET Centre, Department of Radiology, Haukeland University Hospital, N-5021 Bergen, Norway*

[b] *Section for Radiology, Clinical Institute I, University of Bergen, Bergen, Norway*

ARTICLE INFO

Article history:
Received 30 April 2013
Received in revised form
10 December 2013
Accepted 8 January 2014

Keywords:
Web application framework
PHP
CakePHP
Clinical Data Management System
Investigator-initiated clinical research

ABSTRACT

Clinical trials aiming for regulatory approval of a therapeutic agent must be conducted according to Good Clinical Practice (GCP). Clinical Data Management Systems (CDMS) are specialized software solutions geared toward GCP-trials. They are however less suited for data management in small non-GCP research projects. For use in researcher-initiated non-GCP studies, we developed a client–server database application based on the public domain CakePHP framework.

The underlying MySQL database uses a simple data model based on only five data tables. The graphical user interface can be run in any web browser inside the hospital network. Data are validated upon entry. Data contained in external database systems can be imported interactively. Data are automatically anonymized on import, and the key lists identifying the subjects being logged to a restricted part of the database. Data analysis is performed by separate statistics and analysis software connecting to the database via a generic Open Database Connectivity (ODBC) interface. Since its first pilot implementation in 2011, the solution has been applied to seven different clinical research projects covering different clinical problems in different organ systems such as cancer of the thyroid and the prostate glands.

This paper shows how the adoption of a generic web application framework is a feasible, flexible, low-cost, and user-friendly way of managing multidimensional research data in researcher-initiated non-GCP clinical projects.

1. Introduction

Clinical studies in human medicine generate multidimensional data sets with numerous observations that are best administered using dedicated software solutions for data entry and analysis. At our molecular imaging center, we needed a flexible, scalable, and affordable solution for data management in our own researcher-initiated studies.

Clinical Data Management Systems (CDMS) are a family of client–server applications aimed at pharmaceutical trials [1]. Such trials are conducted for regulatory approval of

* *Correspondence to*: Centre for Nuclear Medicine and PET, Department of Radiology, Haukeland University Hospital, Jonas Liesvei, N-5021 Bergen, Norway. Tel.: +47 55977643; fax: +47 55977602.
E-mail addresses: martin.biermann@k1.uib.no, martin.biermann@helse-bergen.no

0169-2607
http://dx.doi.org/10.1016/j.cmpb.2014.01.007

a drug or medical appliance by regulatory bodies such as the Federal Drug Agency (FDA) or the European Medicines Agency (EMEA). Design, conduct, and data management in such trials are governed by stringent international conventions such as Good Clinical Practice (GCP) [2] in addition to national legislation [3]. The design of such trials is invariably prospective, usually randomized, and, if possible, double-blinded, and outcome measures (such as total mortality or disease-related mortality) are set in advance [4]. Documentation must be tamper-proof [2] to avoid potential allegations of fraud as billions of dollars are at stake for the pharmaceutical company that developed the drug and sponsors the trial [5]. Independent contract research organizations (CRO) specialize in running trials in a GCP-compliant manner. These days, data entry will most often be conducted via electronic case report forms (eCRF) using CDMS with an internet portal [6].

Non-commercial, researcher-initiated studies will often follow less formal exploratory designs aimed at gaining new insights into a given problem. At our molecular imaging center, we combine hybrid imaging – single photon emission computed tomography (SPECT) and positron emission tomography (PET) both acquired in conjunction with computed tomography (CT) – with other radiological modalities such as ultrasound (US), magnetic resonance imaging (MRI), and US-guided biopsies both in our clinical routine and in our research projects. This yields complex data sets comparing several imaging modalities (such as US, PET, SPECT, and contrast-enhanced CT) with cytological (US-guided biopsy) and histological (after surgical treatment) verification in one or several tumor lesions in a large number of patients. Projects are often interdisciplinary, involving different clinical specialists (e.g. surgeons and oncologists), imaging specialists (nuclear medicine and/or radiology), and laboratory specialists (pathology, cytology, clinical chemistry) in the scope of a single research project such as multimodal imaging for thyroid cancer [7].

For use in our own non-GCP clinical research projects and based on earlier experience with a custom-designed data management system for a clinical trial [8,9], we were looking for a system that met the following specifications: (1) The system should be network-based, allowing for concurrent data entry by several authenticated users. (2) The system should meet all current regulatory requirements in respect to data protection and security. (3) The system should allow for hierarchical data models supporting complex entity relationships and provide built-in mechanisms to enforce relational integrity. (4) Modifications to the data models must be easy to implement even when data acquisition is under way. (5) The system should be cheap so that it can be shared between groups and projects without being limited by software licensing. (6) The software should be vendor-independent and multi-platform (e.g. Linux, Microsoft Windows®) so that it can be expected to be viable for the entire duration of projects spanning several years [9,10].

Finding no suitable software solution that met all our current requirements, we set out to develop a new simpler and more scalable solution for data management in our own clinical research projects.

2. Related work

Requirements for GCP-compliant CDMS have been reviewed in depth by Ohmann et al. [11]. An overview of available systems is provided by a recent European Survey [1]. At the 74 study centers, 39 different systems were in use in 2008/2009: 18 self-developed proprietary, 17 commercial, and 4 open source. The latter include the increasingly popular OpenClinica (https://community.openclinica.com), which is based on 3-tier architecture with an apache tomcat web application server (http://tomcat.apache.org) with a PostgreSQL (http://www.postgresql.org) database backend.

An alternative approach suited for large non-GCP research projects is the establishment of an integrated information technology (IT) framework where structured data from electronic medical patient records are reused for clinical and translational research based on a single source concept of data entry [12–15]. When interfaced against other systems such as laboratory information systems, such frameworks will not only eliminate duplicate documentation requirements for physicians, but can help improve patient safety by providing on-line surveillance of critical events such as adverse drug reactions (ADR) [16,17]. Since these frameworks heavily rely on the exchange of information between different systems, information is most often expressed using standardized dictionaries, such as WHO-ART for coding ADR or LOINC for using laboratory tests [16,18,19]. Due to their complexity, the establishment of such frameworks requires a major commitment from the health care provider such as major comprehensive cancer centers, limiting their availability and accessibility to the individual researcher. In addition, there is a growing number of web-based solutions for outcome surveillance in a clinical or research setting such as CAISIS (http://www.caisis.org/), OIO (http://sourceforge.net/projects/open-outcomes/), Medintux (http://medintux.org) and FreeMED (http://freemedsoftware.org) as well as mobile solutions for data entry [20].

3. Design considerations

We had previously developed our own client–server application based on an Oracle database (Oracle Corp. Inc., Redwood City/CA) with Oracle Forms graphical clients [8] for data management in a prospective randomized multicenter trial. The MSDS trial on external beam radiotherapy (RTx) for locally advanced differentiated thyroid cancer (DTC) was run in close collaboration with the Department of Biometrics/Competence Centre for Clinical Studies (KKS) at the University of Münster. Challenges in managing the trial were its interdisciplinary design involving endocrine surgery, pathology, radiotherapy, and nuclear medicine with separate reference centers for each specialty, and the trial's size (429 patients), duration (10 years), and geographical distribution (50 participating centers in 3 countries). As none of the then available CDMS were found to be suited to the task within the funding constraints of the trial, we decided to proceed with our own development based

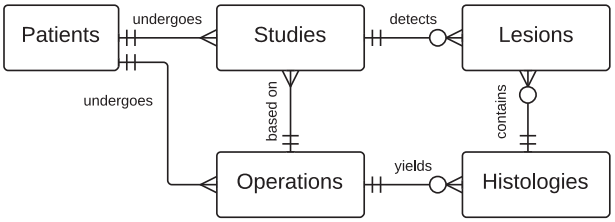

Fig. 1 – Entity relationship diagram describing the data model underlying the application. See text for details.

on earlier prototypes on the same platform. The system was operational between 2000 and 2010 [9].

The client–server architecture had the advantage that data could be entered simultaneously by several concurrent users and that all entered data could be validated by the client application before being committed to the database. A number of fundamental drawbacks became however apparent over time: (1) client application updates were difficult to enforce at distant sites. (2) Each change even in a minor database table needed reprogramming, recompilation and redistribution of the client software. (3) Oracle stopped support for the Oracle Forms platform in 2006. Later it became impossible to install the client application with the module needed for encrypted client server communication, and the Forms client conflicted with other Oracle client installations in our hospital network. (4) The data models were too complicated, as they had to be derived from the paper-based case report forms approved before the start of the trial. (5) Data analysis based on Structured Query Language (SQL) was inflexible. Changes in a single column would have to be propagated through a series of cascading SQL views, making even minor changes costly to implement.

Based on this experience, we set out to develop a new, simpler solution for data management in our own clinical research projects which met the specifications as detailed in Section 1.

Most observations in clinical studies are based on multiway entity relationships (Fig. 1). A patient may have many follow-up visits or imaging studies (hence referred to as "studies") (1:*n* relationship), and each of these studies may generate zero, one, or many findings (hence referred to as "lesions"). All entities may be associated with categorical variables such as disease status or uptake of a contrast agent, or continuous variables such as a physical measurement or the blood level of a biochemical marker. The most appropriate way of handling multidimensional data is a relational database. We thus decided to base our development on a transactional database management system (DMBS) using Structured Query Language (SQL).

To facilitate reliable and consistent data entry, a customized graphical interface with on-screen forms is mandatory. If several users are to take part in data collection, a network-based client–server architecture is necessary. In accordance with modern internet practice, we opted for a three-tier architecture consisting of database server, application server and "thin" clients (Fig. 2). To avoid the need for excessive hand coding of web pages and increase reusability of code, we looked for a web framework that would allow

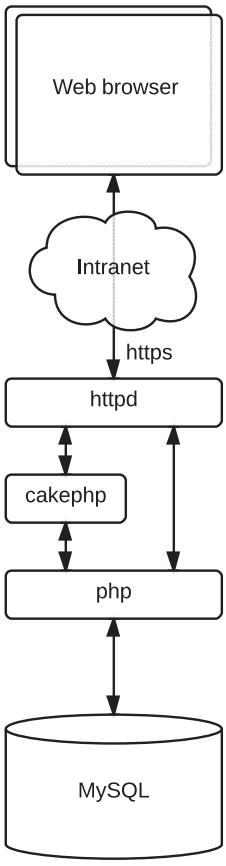

Fig. 2 – System architecture. Thin clients (web browsers) are connected to the Apache/httpd application server running CakePHP/PHP on top of the MySQL database.

the easy generation of a graphical front end for a given SQL database.

4. System description and methods

4.1. *Data models*

To facilitate consistent entry and analysis of the data, data models need to be fully normalized, simple and universally applicable. The data model outlined in Fig. 1 based on only five tables has so far met the demands in all our current projects. Despite being simple, it still respects all the pertinent object–entity relationships in our research data. Limiting the number of tables containing observations greatly streamlines the data analysis as fewer tables need to be joined during data analysis.

4.2. *Implementation technologies and development details*

CakePHP was chosen as an application framework. CakePHP is one of several open source application frameworks such as Ruby, Zend or Symphony [21] that allows the rapid generation of a web-based graphical user interface for an SQL database. CakePHP is written in PHP and distributed under

the MIT License. Like many competing frameworks, CakePHP incorporates a number of key concepts and technologies that reduce the need of hand coding web application pages [22]. The Model View Controller (MCV) paradigm separates the application logic (controller) from the underlying data models (model) and the physical webpages (view). "Convention over Configuration" imposes a set of strict rules on the structure of the underlying SQL tables including full normalization of the underlying database. When these rules are followed, CakePHP will "automagically" [22] choose the correct interface elements to represent a given entity, e.g. a dropdown list for representing categorical data or a checkbox for logical data (see Supplementary Materials 1). In combination with the "natural language" paradigm, this leads to easily maintainable databases with user-friendly human readable uniform resource locators (URL) for the web interface. Rapid development is promoted by scaffolding the application: A complete Create Read Update Delete (CRUD) interface for a table can be generated by 10 lines of CakePHP code. The table is then dynamically read from the database server, and one can make repeated changes to the underlying SQL table without having to re-code the application. When the database meets all requirements, the scaffolded application can be cast into PHP code by running the "bake" script. The static PHP code can then be manually edited to produce the final web-based application. Special functionality not available within the CakePHP framework such as semi-automatic import of patient and study data from one of the department's image databases is implemented outside the framework by means of hand-coded PHP pages. To facilitate re-use of existing code, CakePHP projects can be cloned from existing related projects via a custom developed Python script running on the server, while the underlying MySQL database can be cloned by means of a custom PHP script.

For data analysis, a modular architecture is chosen. First, data are re-aggregated by means of SQL views implemented on the database server. Statistical analysis software is then connected to the database server via Open Database Connectivity (ODBC) for further analysis and for quality control against the original observations. In line with requirement #6, we use the open source statistics program R [23]. The library "RODBC" is used for data import [24]. Compared to library "RMySQL" [25], "RODBC" has the advantage that all character data are automatically converted to factors by default, greatly facilitating statistical analyses in subsets of the data with a minimum of coding (See Supplementary Materials 1 for an example illustrating the complete workflow from scaffolding a CakePHP application, data transformation with MySQL, and data analysis with R).

To restrict access to the database, the server is run inside the protected hospital network. Communication between thin client and application server is encrypted by Transport Layer Security (TLS, https). Each project has its own user administration with usernames and passwords. User roles were implemented through an extension of the CakePHP 1.x framework described in Supplementary Materials 2. Current Norwegian legislation [3] demands that data stored in research databases should not contain patient identification, and that the key list between the patient code and the unique national person identity number (NPID) be stored in a location separate from the other observations. This condition is met by using the DBMS to partition the data set into different databases so that the key list resides in a database that is only accessible to administrators. To avoid duplicate entries in the patient table, a hash of the NPID is retained with the data. NPID and hashed NPID are inserted into the key list in the protected database by means of an SQL trigger (see Supplement 3 for details).

4.3. *Hardware requirements*

The application can be run on any Linux or Windows Apache/MySQL/PHP (LAMP/WAMP) server. The original set of web applications was developed on a LAMP server running on an x586 Intel personal computer (PC) with 2 GB of RAM under 32-bit Open SuSE Linux 11.2 and has recently been moved to Open SuSE 12.3-64. CakePHP 1.3.x was downloaded from http://cakephp.org, and the CakePHP finder plug-in from http://cakedc.com. A second Open SuSE server provides source code version management via subversion (http://subversion.apache.org) and file backup via Bacula 5.x/MySQL (http://bacula.org). For statistical analysis, *R* is run on the Win7-64 desktop via an ODBC-connection to the remote MySQL server using the "RODBC" package [23].

4.4. *Methods for system evaluation*

To analyze changes in the PHP source code over time, the commit logs of the subversion server were pre-processed with statsvn (http://sourceforge.net/projects/statsvn) and then manually analyzed using a custom-designed CakePHP database application and R (see Supplementary Materials 1). For comparison between projects, Fisher's exact test was used for categorical data (types of commit) and Kruskal–Wallis test for not normally distributed numerical data (lines of code per commit) with a significance level of $p < 0.05$ (two-sided).

To assess user experience in an unbiased manner, a user survey comprising 22 questions was conducted using SurveyMonkey (http://www.surveymonkey.net) in July 2013 (Supplementary Materials 4). Survey results were plotted using R library "ggplot2" [26].

To facilitate comparison of the system with competing solutions for data management, SPSS (v. 22.0.0.1, IBM Inc.) was installed on a hospital system under Microsoft Windows 7-64 as an example of a popular statistics program, while an OpenClinica server (v. 3.1.4; https://community.openclinica.com) was set up as an example for a state-of-the-art open-source CDMS. Systems were evaluated by the author (M.B.) by entering test data originating from multimodal imaging of thyroid cancer patients. Criteria included: GCP-compliance, provision of relational integrity, ease of upgrading the application in a networked environment, ease of making changes to the data model (such as adding a table column), and representation of categorical data by means of dropdown lists for rapid and reliable data entry.

To assess current standards for data management and statistics in medical imaging research, the full manuscripts of all original human cancer imaging studies published in the two highest ranked medical imaging journals in the entire year of 2013 were analyzed in respect to data management and

statistical methodology. In 2012, *Radiology* had an impact factor of 6.339, and the *Journal of Nuclear Medicine and Molecular Imaging* (JNMMI) of 5.774. Data were entered into a custom CakePHP database, and analyzed with R. The Kruskal–Wallis test was used to compare the number of human subjects per study (not normally distributed) and Fisher's exact test for comparing the data management and statistics solutions, respectively, between the two journals.

5. Results

5.1. *Clinical research applications*

Since the first prototype was implemented in autumn 2011, we are currently running seven medical imaging-related research projects on our application server. For each clinical project, a dedicated CakePHP application is run as a separate CakePHP project with its own unique base URL and database partition. Usage data on the three major current projects are listed in Table 1.

Table 1 – Performance data on the three sample research projects. Number of main data tables (excluding look-up tables), number of active users included in the survey, number of patients/subjects in the patients table, total number of records in the data tables (excluding the patients table) as of 25 August 2013.

	petdb	mmtc	pro
Project start	11/2011	9/2011	9/2011
N active users	10	2	3
N patients	3708	61	333
N records	7774	372	6387

The first application called "petdb" (PET database) was developed for monitoring all PET examinations performed in our department since the start of clinical PET in April 2009. The basic observation unit is a patient. PET studies are automatically imported from one of the department's image databases through a special hand-coded PHP script on the Apache/PHP application server. After import, diagnoses are assigned to each study according to the International Classification of Diseases (ICD-10) through the web application's

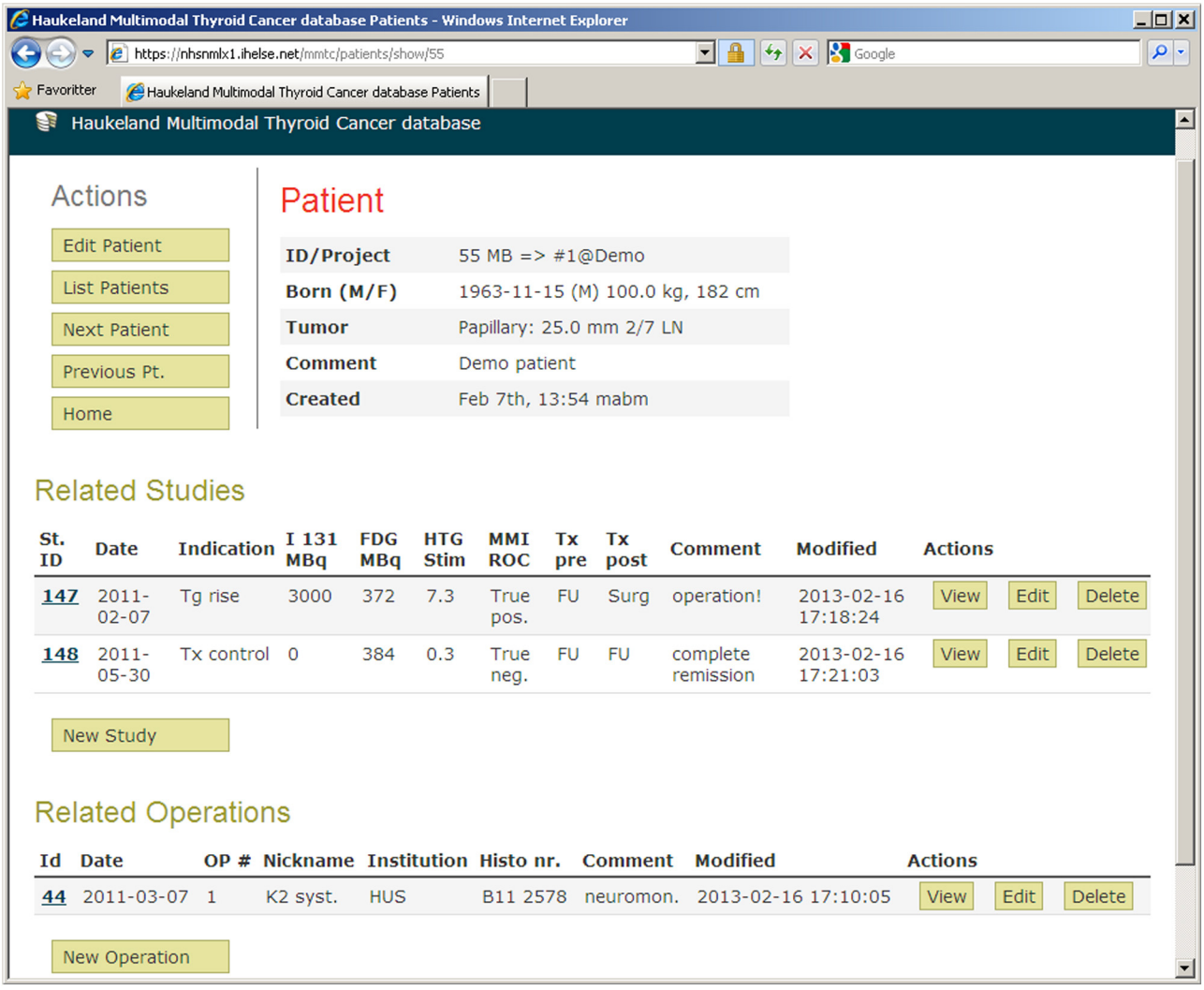

Fig. 3 – The main patient view in the application. A given patient may undergo one or many imaging studies and zero or many operations. The patient shown is fictional. For economy of space, the "view" page in the application has been simplified.

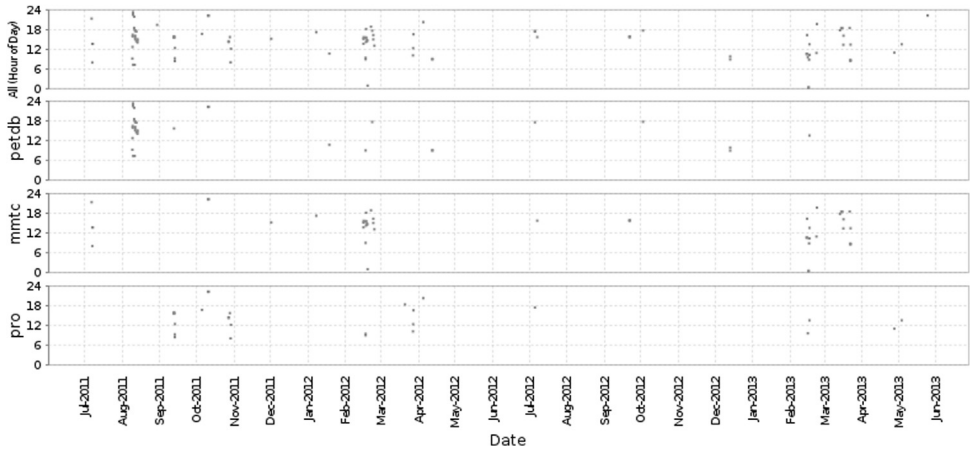

Fig. 4 – Code changes (svn commits) over time between July 2011 and June 2013 according to date (*x*-axis) and time of day (*y*-axis) for all (All) and the 3 individual projects (petdb, mmtc, pro).

graphical interface. The application with currently over 4000 PET-studies has been routinely used in our department since 2011 by more than ten technical as well as non-technical users both for research and for controlling expenditure at the PET-centre. A publication on the clinical use of PET in our health region (Helse Vest) since 2009 is in preparation.

Our second application called "mmtc" (multimodal thyroid cancer) was specifically designed for our on-going study on multimodal imaging of patients with suspected recurrent thyroid cancer [3]. A sample screen is shown in Fig. 3. A patient can have one or more "studies" that comprise several modalities: PET-scanning, contrast-enhanced CT, US pre- and post-PET, as well as US-guided fine needle biopsy. Each "study" can produce zero to many findings called "lesions". Each lesion, be it a local recurrence, tumor spread to a lymph node or a distant organ, or an enlarged lymph node or other benign finding, is registered as one record in the lesions table. Patients can undergo one or more operations, each of which is stored as one record in the "operations" table. Each operation can yield one or more pathological preparations, which the pathologist examines for tumor lesions. Each preparation is stored as one record in the "histologies" table. By assigning links between the lesions and the histologies tables we can answer the question of how many tumor foci found at microscopic examination are missed in medical imaging studies. This application has since been cloned into applications specific to multimodal imaging of hyperparathyroidism (>600 examinations), and endometrial cancer (>100 examinations).

Our third major application called "pro" (Prostate) is dedicated to MRI of the prostate. The data model underlying the lesions table had to be modified to cover several sets of observations (three radiologists who read three MRI series each; histopathological Gleason score by one pathologist) in each of 27 anatomical segments in the prostate gland. The lesions table was expanded to cover all 27 segments. These segments are shown in anatomical arrangement in order to eliminate coding errors by the observers. Each radiologist codes four sets of observations per study (three MR series, one overall impression) while the application blinds him/her as to the pathology and the observations entered by the other radiologists. This application, which currently contains more than 60 000 prostatic segments, has been in use since September 2011. A manuscript has recently been submitted [27].

5.2. *Evolution of code over time*

Code changes over time in the three above projects are plotted in Fig. 4 based on the commit logs of the subversion server. From July 2011, there were 82 committed software versions for the three projects. 27% of the changes were due to changes in the underlying data model (such as extra columns in the main data tables, new types of categorical data), 10% due to changed data validation rules without changes to the database, while

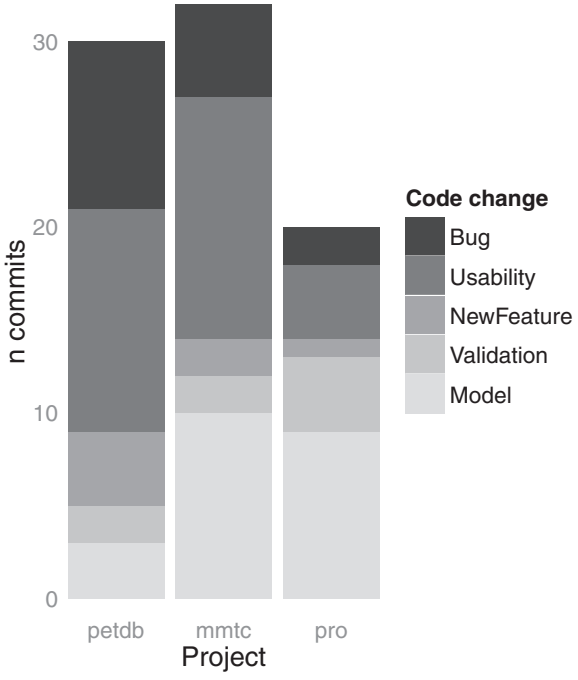

Fig. 5 – Types of code changes (svn commits) in the three projects. See text for details.

35% were due to usability enhancements, 9% to new features, and 20% due to bug fixes (Fig. 5). The median changed lines of code were 11 for model changes, 30 for changes in validation rules, 16 for usability enhancements, 59 for new features, and 9 for bug fixes with a median of 3 source code files affected. There were no statistically significant differences in the types of changes or the number of lines per change between the three projects.

5.3. *User satisfaction survey*

A user satisfaction survey was conducted among the 14 active users of the software (excluding the developer M.B.), all of whom responded. 57% of users were over 40 years old with an even male to female ratio. 29% had college-level education (technician, mercantile), 71% university training, 22% at PhD-level. 72% of users characterized themselves as "normal" computer users, 1 as computer novice, 2 as power users, and none as IT professional or developer while 1 user did not disclose her level of expertise. 57% of the users had been using the software for more than 1 year. 78% of users had been using the software for one research project, 22% for two. 93% of users had been involved in data entry while 21% used the platform for publications and abstracts with a total of 2 manuscript submissions so far. 56% of users had edited datasets belonging to more than 100 study subjects. System downtime reported by the users was nil. Four users reported experiencing bugs, and 1 user missing features in the software which interfered with their work up to 3 times a year. All bugs were repaired within 24 h while missing features were typically implemented within one week. When asked what they liked best with the software, 8 out of 10 users emphasized the software's user-friendliness. Average score was 4.7 on a 5-point scale from 1 (worst) to 5 (best) (Fig. 6). 79% of the users would like to use the software for future projects.

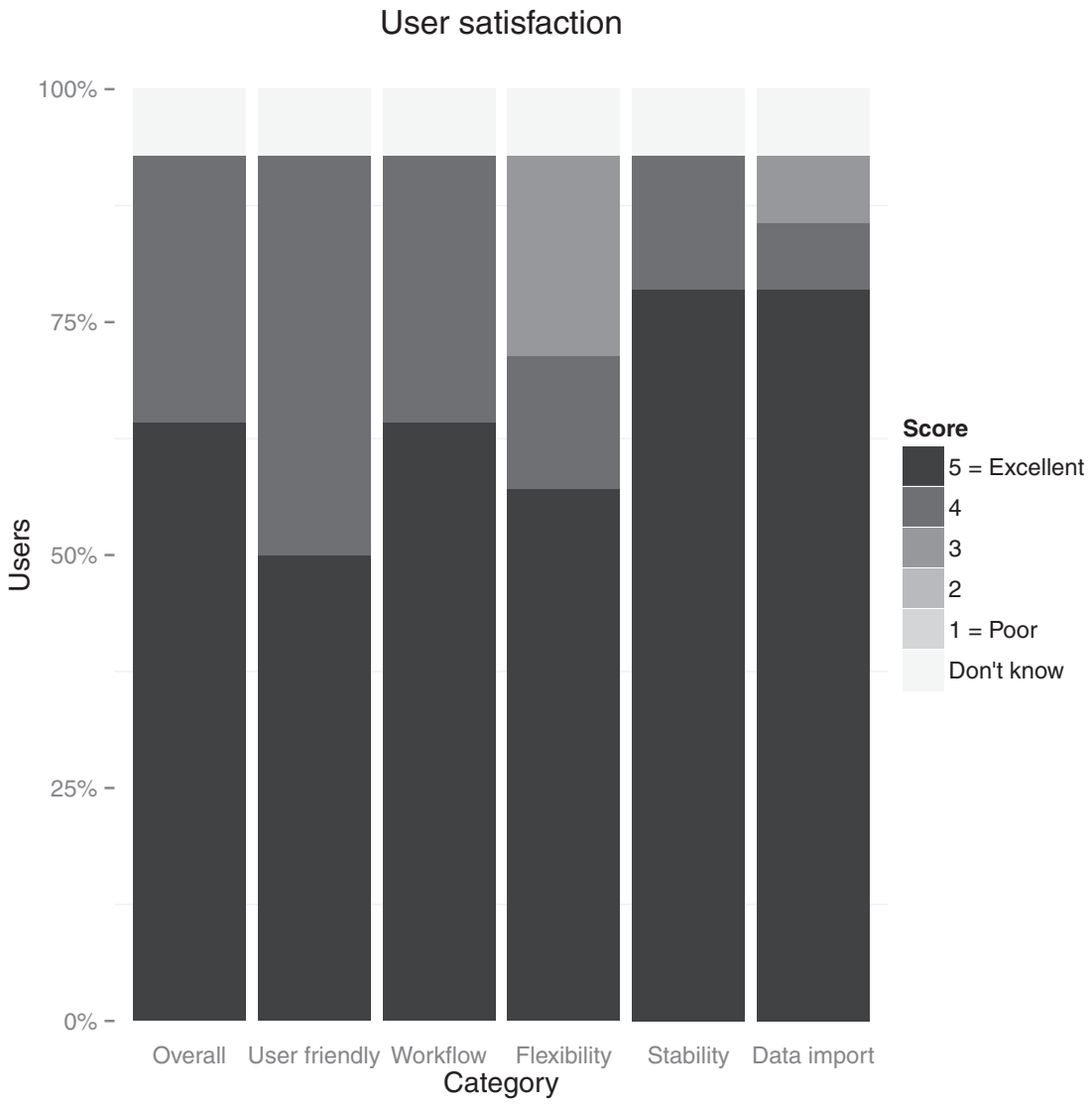

Fig. 6 – User satisfaction scores for $n = 14$ active users of the software. One user who used the software for report generation only did not report scores.

Table 2 – Feature comparison between different platforms for data entry.

Feature	SPSS	MSDS database	CakePHP	OpenClinica
Architecture	Single user	2-tier	3-tier	3-tier
Public domain	–	–	+	+
Good Clinical Practice (GCP) compliance	–	(+)	–	+
Database engine	Flat file	Oracle 9i	MySQL 5.x	PostgreSQL 8.x
Relational integrity	–	+	+	+
Application server	–	–	Apache/PHP	Apache tomcat
Client software	–	Oracle Forms	HTTP browser	HTTP browser
Ease of upgrading application		–	++	+
Ease of adding table columns	++	(+)	++	–
Dropdown lists for categorical data	+	+	+	+

Table 3 – Data management in current medical imaging research. Original cancer imaging research articles in human subjects published in Radiology and Journal of Nuclear Medicine and Molecular imaging (JNMMI) in the year of 2003. See manuscript for details. N subjects: median number of subjects per study, range. Percentages refer to the total number of original research papers related to cancer-imaging; 23 articles used more than one statistics program.

Journal	Radiology	JNMMI	
Original research papers in humans	309	141	
Cancer imaging papers	92	62	
N subjects	98 [10; 688,481]	41 [4; 286]	$p < 0.05$
Dedicated data management	5 (5%)	0 (0%)	n.s.
Statistics software			
Not mentioned	19 (21%)	24 (39%)	$p < 0.05$
SPSS	26 (28%)	24 (39%)	
SAS	24 (29%)	3 (5%)	
R	8 (9%)	4 (5%)	$p < 0.05$
STATA	5 (5%)	2 (3%)	
others	25 (27%)	14 (23%)	

5.4. *Comparison with other systems*

A feature comparison between our CakePHP solution with other popular platforms for data entry is listed in Table 2.

5.5. *Current research methodology in medical imaging*

Only five of 154 cancer-related original research articles in human subjects published in the two leading medical imaging journals in 2013 claimed the use of dedicated solutions for data entry (see Table 3): four mammography screening studies and one registry study. None used a CDMS for data management. The most popular statistics program was SPSS, followed by SAS and R, while 21% of the articles in Radiology and 39% in JNMMI did not specify which statistics program was used.

6. Discussion

Outside the sphere of GCP-compliant clinical trials run for approval by regulatory bodies, data management appears to be an often-underappreciated topic in clinical research. While a survey conducted in 2009 among over 70 European academic centers running clinical trials found that 90% had CDMS in routine use [1], the vast majority researcher-initiated non-GCP studies are restricted to spreadsheet software [28] or statistics programs for data collection. These suffer from a simple tabular representation of the data and from being single-user systems. There is no good reason why standards for data consistency and data security in non-GCP researcher-initiated studies should be systematically lower than in clinical trials.

Since the advent of the GCP-standard for clinical trials in 1996, electronic data capture solutions have evolved which meet most, if not all, requirements for usability, scalability, data security and auditing [11]. The most recent of these systems use a 3-tier client–server architecture with database server, application server, and a web browser as the client component. The latter greatly reduces costs for deployment and certification. While most CDMS are proprietary, the proportion of public domain systems is increasing [1,29]. Among the latter, OpenClinica enjoys increasing popularity. As an open source 3-tier client–server system, it meets all the requirements listed in the introduction of this manuscript except feature #4, the easy modification of the data model in an ongoing project. This limitation is however a central feature of GCP, which is based on the concept of a purely prospective clinical trial design with pre-determined outcome measurements. Revisions of a Case Report Form (CRF) and its underlying data model must be difficult to implement.

The life cycle of most researcher-initiated projects, especially when they are of a more exploratory nature, is different. Software applications for research projects are special in that they are often specific to a particular project with very few users and that the life cycle of the application is strictly determined by the duration of the research project. Reusability and robustness of the code are therefore paramount to minimize development costs. As Figs. 4 and 5 document, there were regular code changes in all of our three pilot projects over the entire duration of each project, 37% of them because of adjustments to data model and/or data validation rules. Based on our previous experience with our own custom-developed CDMS for a clinical trial [8], we early on decided that CDMS were

unsuited for managing data in our current projects and looked toward a generic public domain web application framework for our software development.

The main novelty in this manuscript is that we present a simple and flexible approach for data management in researcher-initiated projects based on common public domain components. So far, our project-specific applications have been easy to clone and adapt to use in new research projects focusing on different organ systems and/or research questions, and we now intend to migrate the application for the management of our pre-clinical imaging projects and our departmental pediatric hip imaging registry [30]. The user survey documents the validity of our solution in the context of our three pilot projects. Satisfaction scores among the 14 active users of the software were high (Fig. 6) independent of the level of education or computing experience. Interestingly, the category "flexibility" received the lowest satisfaction scores in the survey. This is presumably because the flexibility of the software lies in the implementation of data models and data analysis, not so much in the user interface, which is designed to enforce a standard workflow for data entry.

The choice of MySQL, CakePHP and R for the engineering implementation is arbitrary, as many other public domain tools such as PostgreSQL (http://www.postgresql.org; supported by CakePHP 1.3.x and 2.x), Ruby on Rails (http://rubyonrails.org/), and Python (http://www.python.org/) have similar functionality. While a full-scale comparison of competing frameworks [21] is beyond the scope of this article, there are however important distinctions from the developer's point of view, which will govern the choice of framework in a given setting: (1) the language of the framework (e.g. PHP versus Ruby or Java). (2) Whether the language is interpreted or compiled. Changes in PHP scripts on a running Apache/PHP server are instantaneously active while code on a Java-based application server needs to be recompiled. (3) Platform dependence. CakePHP runs on any platform that supports an Apache/PHP server, i.e. Linux, Microsoft Windows and Apple MacOS X. (4) Rapid development tools for the dynamic generation of a graphic interface for a given database table. CakePHP provides this functionality through scaffolding. (5) Availability of debugging tools and coding aids such as an integrated development environment (IDE). Debugging tools were lacking in CakePHP 1.x and are greatly improved in 2.x. There is still no native IDE support for CakePHP even though Eclipse (http://www.eclipse.org) and Komodo IDE (http://www.activestate.com) are both good general-purpose PHP editors for CakePHP projects. (6) The direction of the design process. A CakePHP project starts with the design of the database, while other platforms such as OpenClinica start with the interface and let the system create the database. Since much time in the life cycle of a project is spent in the analysis phase, the first approach, which leads to the simplest database structure, is preferable.

Limitations of the system: (1) The system is not intended for conducting clinical trials according to GCP standard. (2) The system is not meant to compete with clinical data warehousing solutions integrated into electronic medical records [12,16,17]. The system is meant to provide a means of consistent data entry where such systems are not available or where the needs for data analysis goes beyond the level of detail provided by such systems. (3) The emphasis of the present system is on simplicity and flexibility with ready adaptability of existing code to new research problems. There is less focus on consistency of data models and data dictionaries across research applications. This flexibility is an advantage when conducting exploratory research projects. For example, there is yet no LOINC term for human thyroglobulin analyzed in the washout of an US-guided fine-needle biopsy [18], a method which we routinely employ in our "mmtc" project. However, the openness of the system entails that existing classifications such as ICD-10 can be readily integrated into the system as for example in our "petdb" project. (4) While the proposed system has been used in single workstation configurations (all 3 tiers on a Windows 8-64 laptop computer) and with up to 15 active users inside the protected hospital network, a major development effort would be needed before the system can be exposed to a larger circle of users and/or less secure networks. The entire application would need to be hard-coded, not just scaffolded, user roles and privileges would need to be more granular, and an audit log would need to be implemented for recording all changes made to the data. While all these changes are possible to implement, they would detract from the main virtue of the system, its simplicity. (5) CakePHP may not have the necessary performance for supporting a very large number of concurrently logged on users.

Methodological limitations: (1) The CakePHP framework is probably only one of several competing application frameworks that are suited to the research applications under discussion. However, a formal comparison between frameworks [21] is beyond the scope of this article. (2) The user survey demonstrates the usability of the present system, but does not provide a comparison between competing systems.

These limitations do however not affect the main conclusion of the paper that the application of a generic web application framework based on the MCV paradigm is a feasible, flexible, low-cost, and user-friendly way of managing multidimensional research data in researcher-initiated studies.

7. Mode of availability of the system or program

A tarball of a sample CakePHP web application can be requested from the author.

Conflicts of interest

None.

Acknowledgements

I thank Dr. Achim Heinecke at the Institute of Biometrics and Clinical Research, University of Münster, Münster/Germany, Prof. Stefan Bruckner, Visualization Group, Department of Informatics, University of Bergen, Bergen/Norway, Assoc. Prof. Albrecht Schmidt, European Space Agency, Madrid/Spain, Prof. Arvid Lundervold, Neuroinformatics Laboratory, Institute of Biomedicine, University of Bergen, Prof. Karen

Rosendahl, Department of Radiology, Haukeland University Hospital/University of Bergen, and Henning Langen Stokmo, M.D., PET-center Bergen, for valuable advice in writing this manuscript. This work was partially supported by the Western Norwegian Health Care (project number 911595) and the MedViz Research Cluster (http://medviz.uib.no).

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.cmpb.2014.01.007.

REFERENCES

[1] W. Kuchinke, C. Ohmann, Q. Yang, et al., Heterogeneity prevails: the state of clinical trial data management in Europe—results of a survey of ECRIN centres, Trials 11 (2010) 79.
[2] International Conference on Harmonization, Good Clinical Practice: Consolidated Guideline. E6 (R1) (1996), http://www.ich.org/fileadmin/Public_Web_Site/ICH_Products/Guidelines/Efficacy/E6_R1/Step4/E6_R1__Guideline.pdf
[3] Federal Drug Administration, Running Clinical Trials: Regulations, 2013, http://www.fda.gov/scienceresearch/specialtopics/runningclinicaltrials/ucm155713.htm
[4] L.M. Friedman, Fundamentals of Clinical Trials, 4th ed., Springer, New York, 2010.
[5] G.W. Williams, The other side of clinical trial monitoring; assuring data quality and procedural adherence, Clin. Trials 3 (2006) 530–537.
[6] Y. Zhang, W. Sun, E.M. Gutchell, et al., QAIT: a quality assurance issue tracking tool to facilitate the improvement of clinical data quality, Comput. Methods Programs Biomed. 109 (2013) 86–91.
[7] M. Biermann, B.C. Reitan, B. Johnsen, et al., False positive FDG-uptake in neck and mediastinum in recurrent differentiated thyroid cancer (DTC), J. Nucl. Med. 53 (Suppl. 1) (2012) 499 (abstract).
[8] M. Biermann, O. Schober, GCP-compliant management of the Multicentric Study Differentiated Thyroid Carcinoma (MSDS) with a relational database under Oracle 8i, Inf. Biometrie Epidemiol. Med. Biol. 33 (2002) 441–459.
[9] M. Biermann, M. Pixberg, B. Riemann, et al., Clinical outcomes of adjuvant external-beam radiotherapy for differentiated thyroid cancer—results after 874 patient-years of follow-up in the MSDS-trial, Nuklearmedizin 48 (2009) 89–98.
[10] L. Zhang, M. Hub, S. Mang, et al., Software for quantitative analysis of radiotherapy: overview, requirement analysis and design solutions, Comput. Methods Programs Biomed. 110 (2013) 528–537.
[11] C. Ohmann, W. Kuchinke, S. Canham, et al., Standard requirements for GCP-compliant data management in multinational clinical trials, Trials 12 (2011) 85.
[12] H.-U. Prokosch, M. Ries, A. Beyer, et al., IT infrastructure components to support clinical care and translational research projects in a comprehensive cancer center, Stud. Health Technol. Inform. 169 (2011) 892–896.
[13] V. Slavov, P. Rao, S. Paturi, et al., A new tool for sharing and querying of clinical documents modeled using HL7 Version 3 standard, Comput. Methods Programs Biomed. 112 (2013) 529–552.
[14] R.S. Santos, S.M.F. Malheiros, S. Cavalheiro, et al., A data mining system for providing analytical information on brain tumors to public health decision makers, Comput. Methods Programs Biomed. 109 (2013) 269–282.
[15] C. Ou-Yang, S. Agustianty, H.-C. Wang, Developing a data mining approach to investigate association between physician prescription and patient outcome—a study on re-hospitalization in Stevens–Johnson Syndrome, Comput. Methods Programs Biomed. 112 (2013) 84–91.
[16] A. Neubert, H. Dormann, H.-U. Prokosch, et al., E-pharmacovigilance: development and implementation of a computable knowledge base to identify adverse drug reactions, Br. J. Clin. Pharmacol. 76 (Suppl. 1) (2013) 69–77.
[17] J.C. Niland, T. Stiller, J. Neat, et al., Improving patient safety via automated laboratory-based adverse event grading, J. Am. Med. Inform. Assoc. 19 (2012) 111–115.
[18] The Regenstrief Institute, Logical Observation Identifiers Names and Codes (LOINC®) Users' Guide, 2013, http://loinc.org
[19] The Uppsala Monitoring Centre, The WHO Adverse Reaction Terminology—WHO-ART, 2005, http://www.umc-products.com/graphics/3149.pdf
[20] J. Meyer, D. Fredrich, J. Piegsa, et al., A mobile and asynchronous electronic data capture system for epidemiologic studies, Comput. Methods Programs Biomed. 110 (2013) 369–379.
[21] B. Porebski, K. Przystalski, L. Nowak, Building PHP Applications With Symfony, CakePHP, and Zend framework, Wiley Pub., Indianapolis, IN, 2011.
[22] D. Golding, Beginning CakePHP from Novice to Professional, Apress, Berkeley, CA/New York, 2008.
[23] R Development Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2012 http://www.r-project.org
[24] B. Ripley, M. Lapsley, RODBC. ODBC Database Access, 2013 http://cran.r-project.org/web/packages/RODBC
[25] D.A. James, S. DebRoy, RMySQL. R Interface to the MySQL Database, 2012 http://cran.r-project.org/web/packages/RMySQL
[26] H. Wickham, Ggplot2 Elegant Graphics for Data Analysis, Springer, Dordrecht/New York, 2009.
[27] L.R. Reisæter, J.J. Fütterer, O.J. Halvorsen, et al., 1.5 T multiparametric MRI using PI-RADS, a zone by zone analysis to localize the index-tumor of prostate cancer in patients undergoing prostatectomy, Acta Radiol. (2013) (re-submitted with changes).
[28] A. Afshar-Oromieh, C.M. Zechmann, A. Malcher, et al., Comparison of PET imaging with a (68)Ga-labelled PSMA ligand and (18)F-choline-based PET/CT for the diagnosis of recurrent prostate cancer, Eur. J. Nucl. Med. Mol. Imaging 41 (2014) 11–20.
[29] G.W. Fegan, T.A. Lang, Could an open-source clinical trial data-management system be what we have all been looking for? PLoS Med. 5 (2008) e6.
[30] L.B. Laborie, I.Ø. Engesæter, T.G. Lehmann, et al., Screening strategies for hip dysplasia: long-term outcome of a randomized controlled trial, Pediatrics 132 (2013) 492–501.